

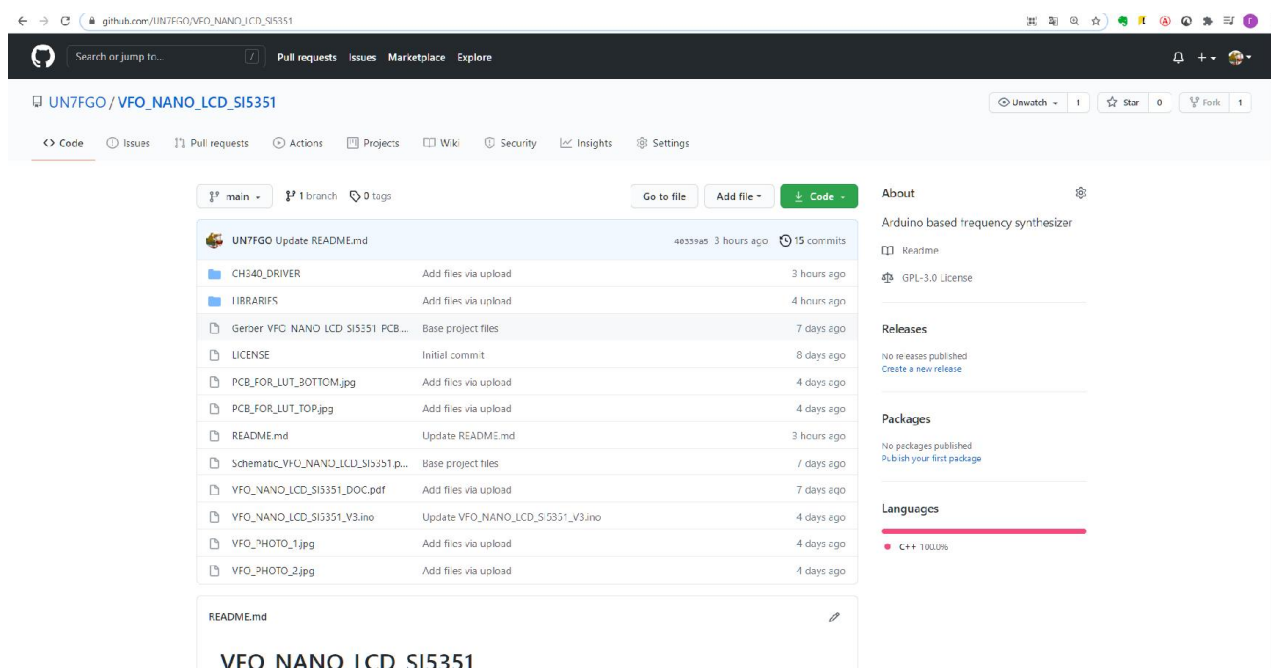
Краткое руководство по загрузке программ в Arduino

предназначено для радиолюбителей «первый раз» видящих, что такое Arduino
на примере проекта VFO_NANO_LCD_SI5351

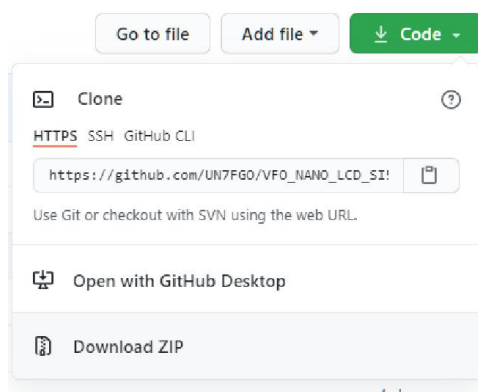
Этап 1. Забираем проект с GitHub.

Переходим на основную страницу проекта -

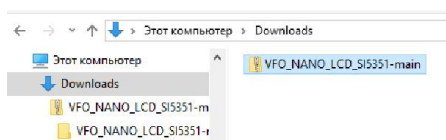
https://github.com/UN7FGO/VFO_NANO_LCD_SI5351 .



Нажимаем зеленую кнопку «Code» и в развернувшемся меню выбираем пункт «Download ZIP».

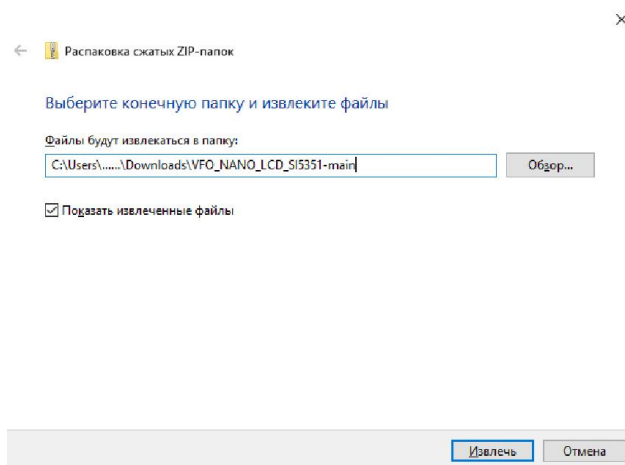


Поздравляю, вы стали обладателем локальной копии проекта.



https://github.com/UN7FGO/VFO_NANO_LCD_SI5351

Осталось распаковать этот архив, для дальнейшей работы с ним. Для этого нажимаем на архиве правой кнопкой мыши и выбираем пункт «Извлечь все...». Появится диалоговое окно.



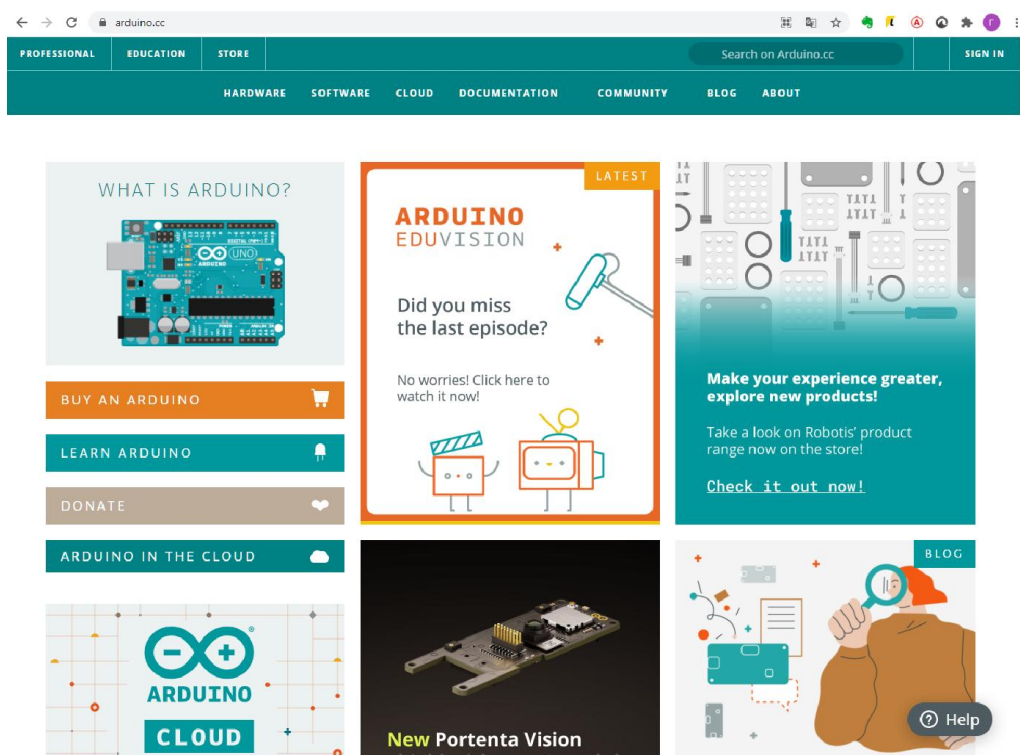
Нажимаем кнопку «Извлечь».

На этом пока оставляем проект в распакованном виде и переходим к подготовке компьютера для работы с Arduino.

Этап 2. Разворачиваем среду для программирования Arduino.

Можно конечно обойтись и без среды разработки приложений для Arduino, но в этом случае вам понадобится еще и программатор UsbAsp и программа в двоичном виде. Так же пропадет возможность хоть как-то скорректировать программу для Arduino под свои нужды. Ну а так как в моем проекте «прошивка» предоставляется в виде исходного кода, который разрешается менять, то мы пойдем по пути разворачивания универсального варианта, который в дальнейшем вполне может пригодиться в других проектах. К тому же среда позволит записать программу не только в Arduino Nano (как в моем проекте), но и в любой другой микроконтроллер серии Arduino, который будет у вас в наличии.

Среда разработки распространяется бесплатно и для ее получения, нам необходимо перейти на сайт проекта, по ссылке <https://www.arduino.cc/>.



Внешний вид сайта может со временем меняться, но разделы вряд ли поменяют свое название.

Нам нужно кликнуть мышкой на разделе меню **SOFTWARE** (вверху страницы).

Arduino Web Editor

Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.

[CODE ONLINE](#) [GETTING STARTED](#)

Downloads

Arduino IDE 1.8.15

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation Instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file

Windows app Win 8.1 or 10 [Get](#)

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

[Release Notes](#) [Checksums](#) (sha512)

[Hourly Builds](#)

Download a **preview of the incoming release**

[Previous Releases](#)

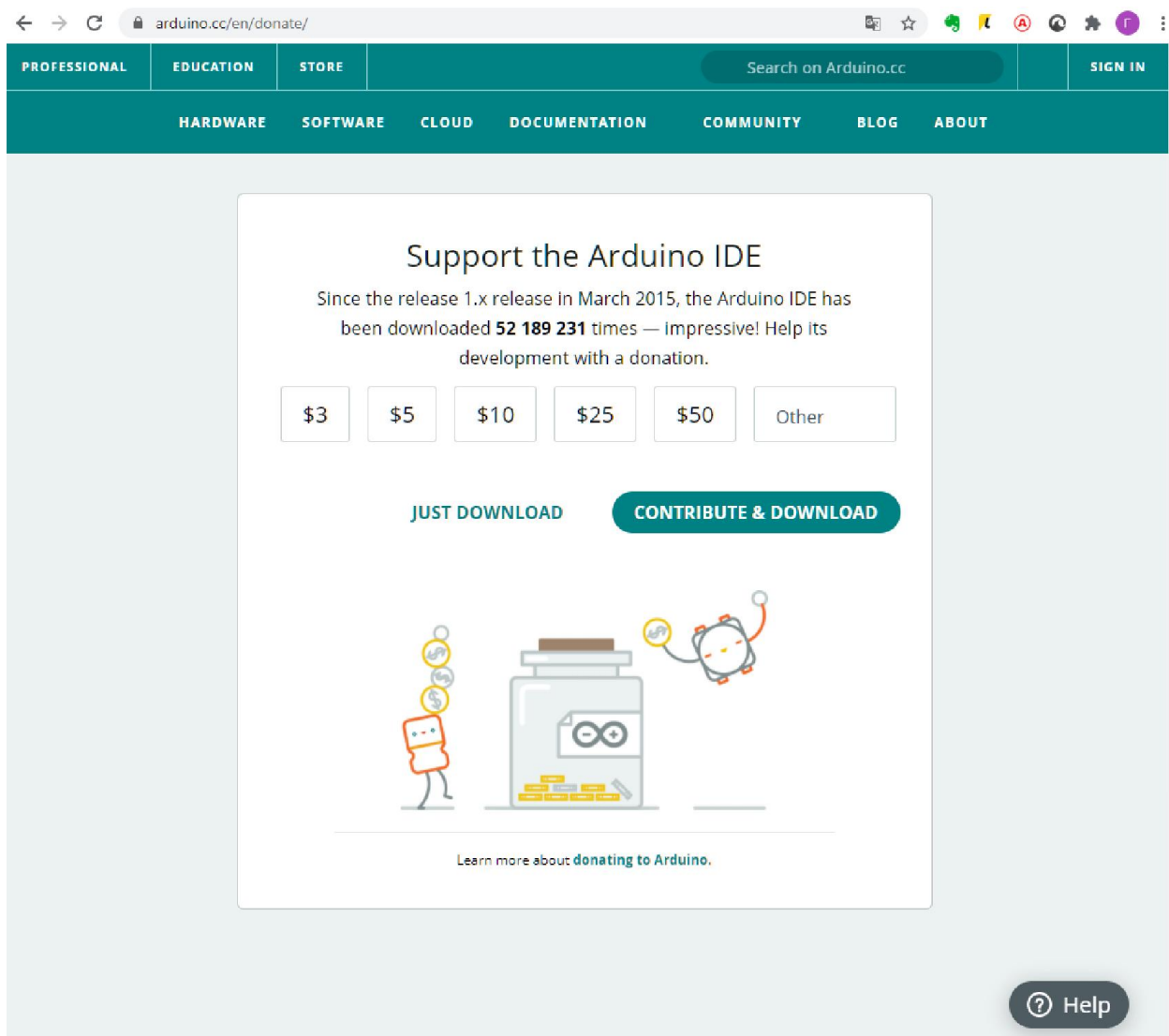
Download the previous version of the current

[Help](#)

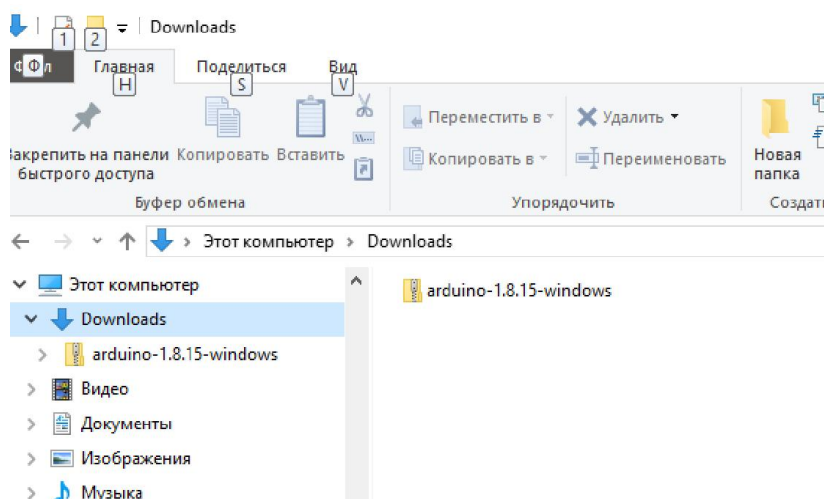
На момент написания инструкции, актуальной версией является 1.8.15. Справа, в темном прямоугольнике предоставлены варианты программы для различных версий операционных систем. Поскольку я предполагаю, что читатель является новичком в области программирования, то вероятнее всего на вашем компьютере установлена операционная система Windows. Вот этот вариант действий я и буду рассматривать.

Из двух вариантов программы для Windows, я предлагаю воспользоваться ZIP-архивом (вторая ссылка). Для начинающих я не рекомендую использовать установочный файл, поскольку при установке «по умолчанию», программа будет установлена в папку «Program Files», а служебные каталоги разместятся в папке профиля пользователя, что усложнит в дальнейшем работу с программой для не очень опытного пользователя.

После выбора пункта «**Windows** ZIP file», нам предложат поддержать проект финансово.



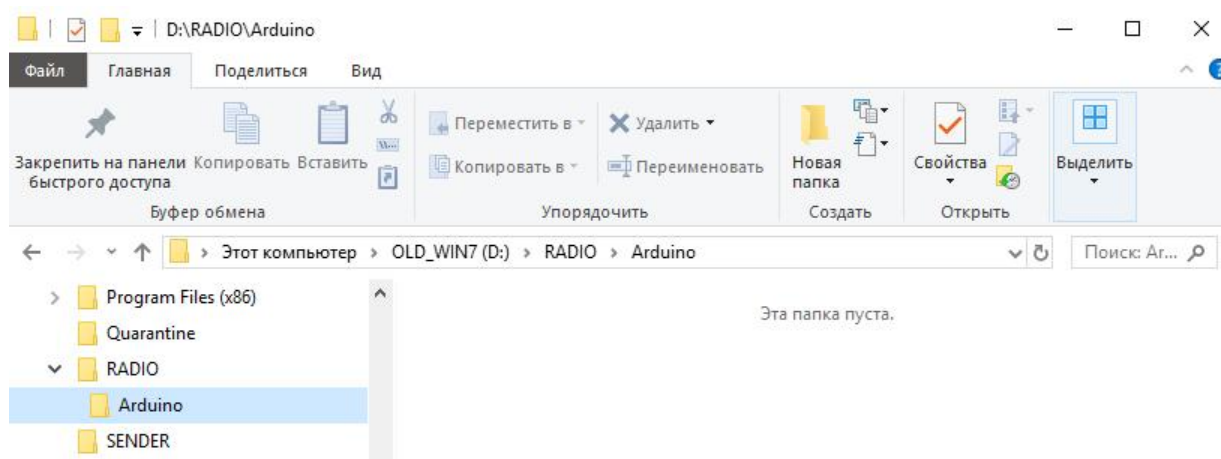
Это не является обязательным действием и если вы не планируете прямо сейчас оказать финансовую помощь проекту, то просто выберите пункт «JUST DOWNLOAD». Начнется скачивание архива. Примерный размер архива 200 мегабайт, поэтому в зависимости от скорости вашего соединения, необходимо будет подождать. В результате, у вас в папке «Загрузки» (Downloads) окажется архив с необходимыми файлами.



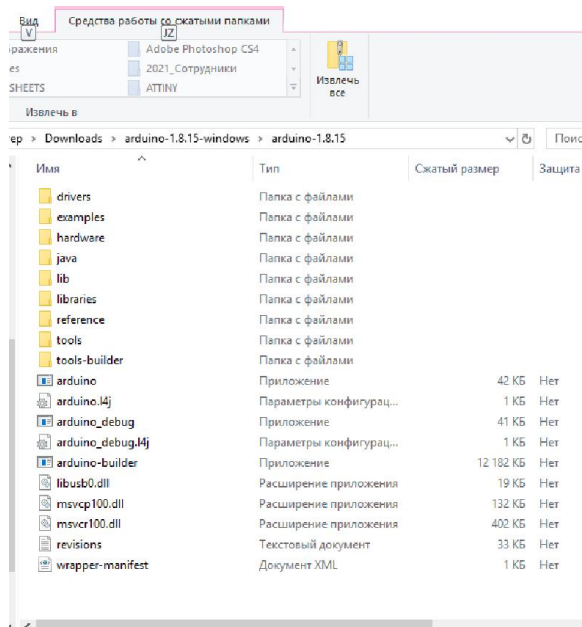
Теперь нам необходимо «правильно» распаковать этот архив.

У вас наверняка есть папка, где вы складываете свои радилюбительские программы. Ну а если такой у вас пока нет, то мы ее заведем.

Некоторые программы очень не любят, когда в имени папки, в которой они лежат, есть символы пробелов или не латинские буквы (например, русские), поэтому мы назовем нашу папку «RADIO» и разместим ее в корневом каталоге диска. Хорошо, если это будет не системный диск C:, а какой-либо другой, поскольку это убережет нас от потери наших программ при переустановке Windows. В новой папке создадим папку «Arduino». Должно получиться примерно вот так:

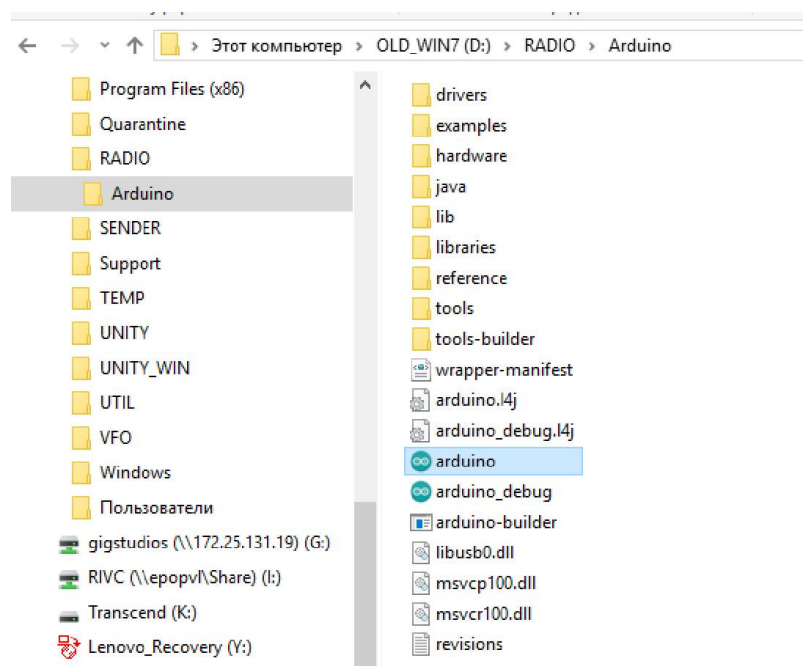


Теперь открываем в «Проводнике» скачанный ранее архив и добираемся в нем до файлов программы.



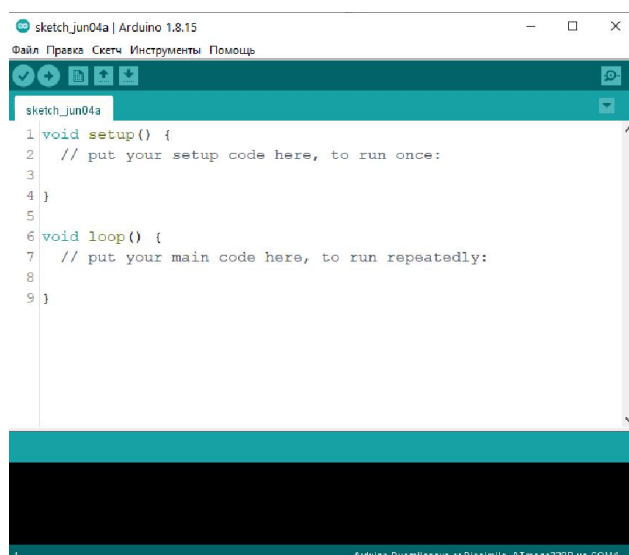
И просто мышкой выделяем перетаскиваем эти файлы и папки в нашу новую папку «RADIO\Arduino». Придется немного подождать, пока архиватор распакует архив и положит файлы на новое место.

В результате мы должны получить полный набор файлов в нашей рабочей папке.



В дальнейшем, для нас будет представлять интерес папка «libraries», поскольку в ней будут располагаться библиотеки, необходимые для работы Arduino с различными внешними модулями или устройствами.

Основной программой, является исполняемый файл «arduino.exe». Давайте попробуем его запустить, чтобы убедиться, что среда разработки у нас находится в работоспособном состоянии. В результате вы должны увидеть примерно вот такое окно приложения.



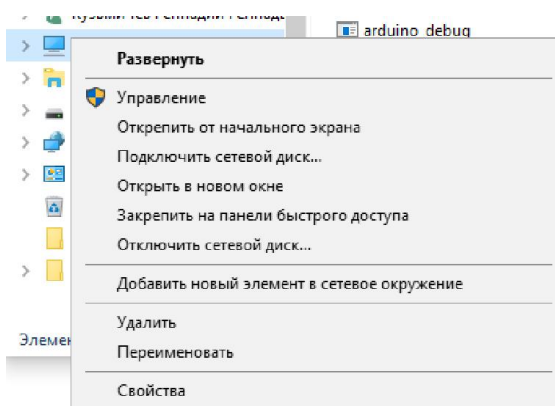
Поздравляю, на вашем компьютере теперь можно полноценно разрабатывать программы для Arduino.

Этап 3. Подключаем Arduino к компьютеру.

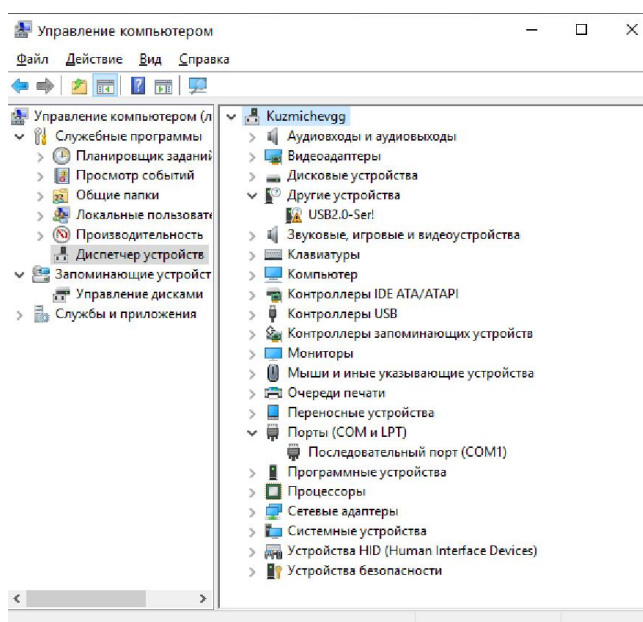
При всей кажущейся простоте, этот этап для многих начинающих становится камнем преткновения и не добавляет желания заниматься Arduino в дальнейшем. Поэтому попробую описать все не явные проблемы этого простого этапа.

Для подключения Arduino к компьютеру обычно используется USB-кабель, с разъемом, соответствующим формату платы микроконтроллера. Отключать и повторно подключать кабель желательно со стороны компьютера, поскольку разъем на микроконтроллере может не выдержать многократного силового воздействия.

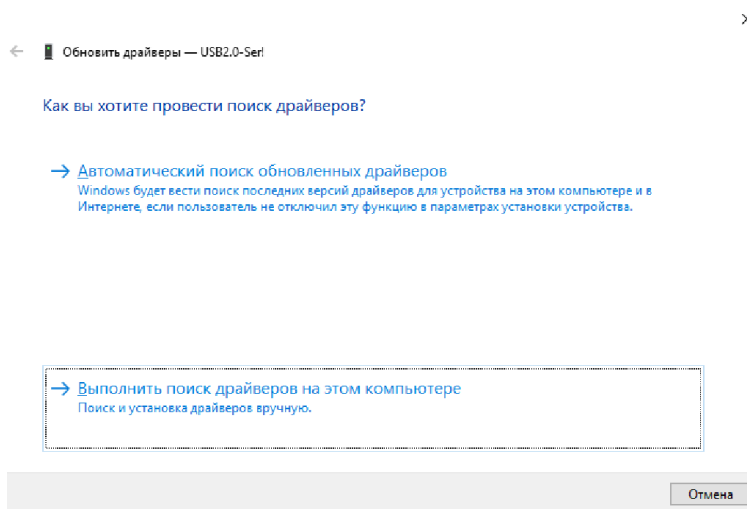
После подключения кабеля от Arduino к компьютеру, на микроконтроллере должен засветиться светодиод, сигнализирующий о наличии напряжения и готовности к работе. Теперь вам необходимо определиться, как же ваш компьютер увидел Arduino. Для этого, на значке «Мой компьютер» (Этот компьютер), который можно найти на рабочем столе или в Проводнике, нажимаем правую кнопку мыши, с целью получить меню.



Нас интересует пункт «Управление». В списке инструментов слева, выбираем «Диспетчер устройств».

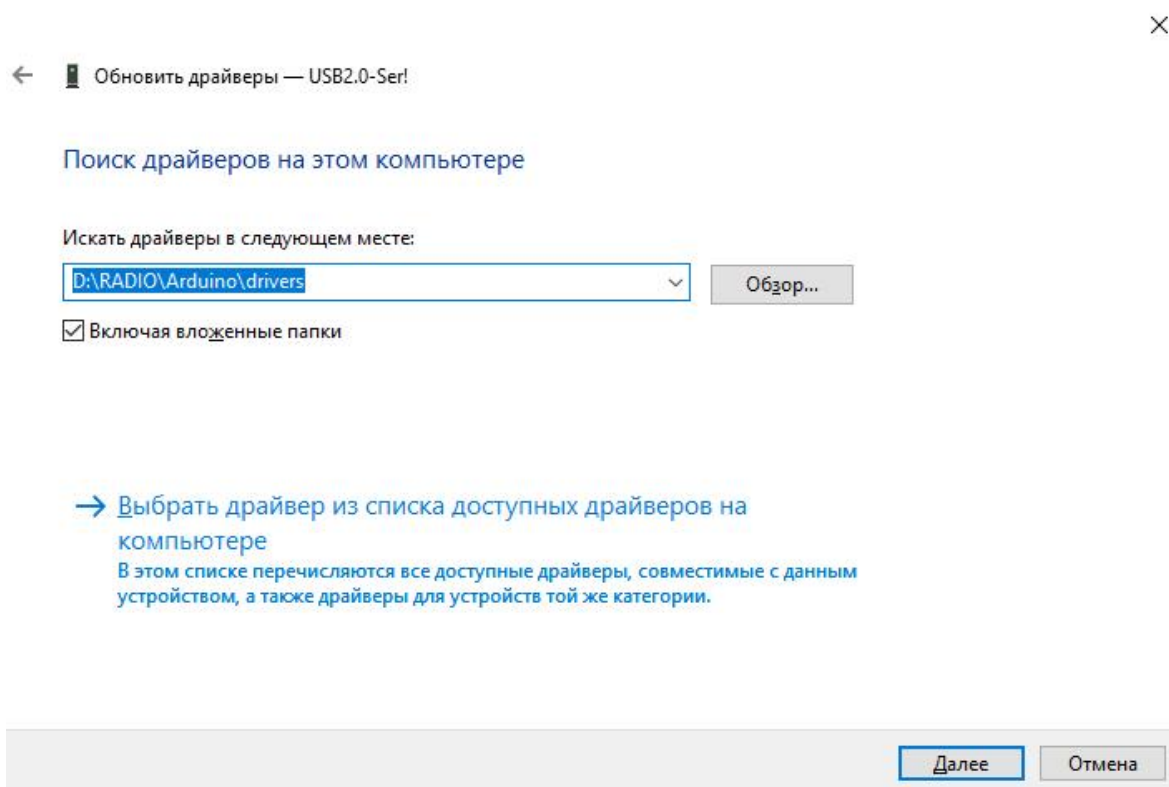


Как видим, у нас есть устройство со странным названием «USB2.0-Ser!», на который система не смогла установить драйвер. Для корректного использования нашей платы Arduino, нужно установить для нее драйвер. Для этого, нажимаем правой кнопкой мыши на этом устройстве и выбираем пункт «Обновить драйвер».



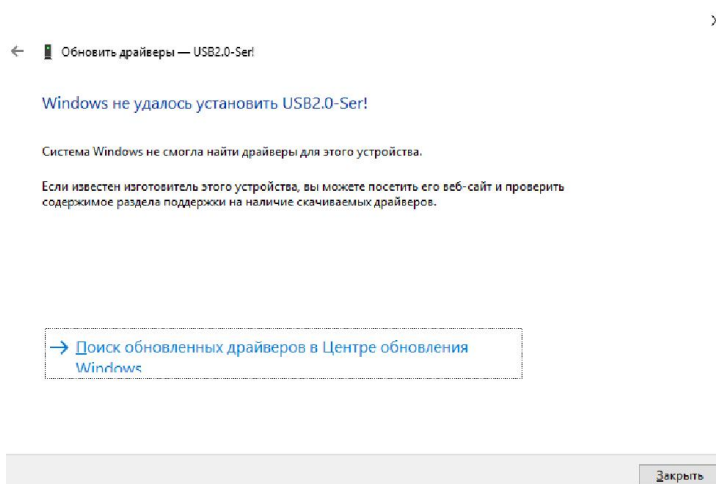
Ну и чтобы не пускать процесс на самотек, выбираем нижний пункт «Выполнить поиск драйверов на этом компьютере».

В комплекте со средой программирования Arduino, уже есть большинство необходимых драйверов, поэтому мы и отправим систему искать драйвер в этой папке, у меня она называется «D:\RADIO\Arduino\drivers».

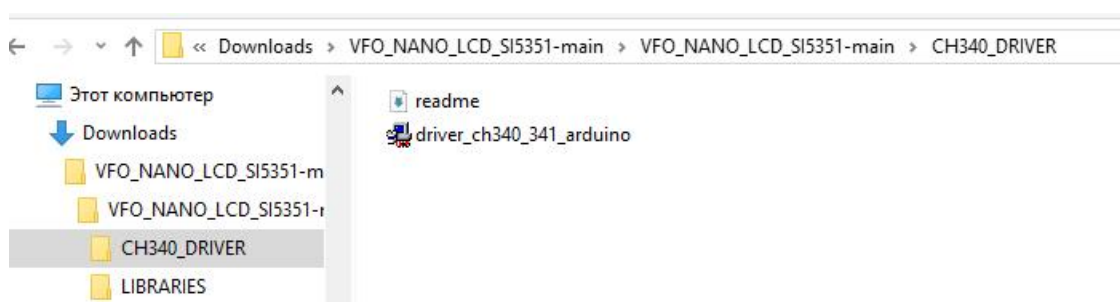


Ну и нажимаем кнопку «Далее».

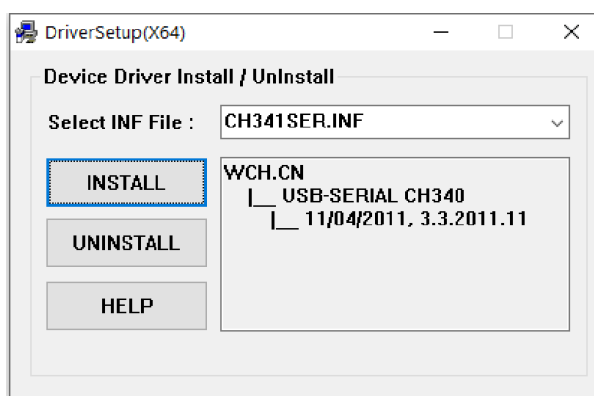
Может получиться так, что системе не удалось подобрать подходящий драйвер.



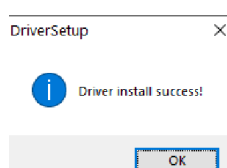
Возможно у вас китайский клон платы Arduino, с микросхемой последовательного интерфейса CH340/341. В этом случае закрываем окно обновления драйвера и открываем папку со скачанным проектом. В ней находим папку «\VFO_NANO_LCD_SI5351-main\CH340_DRIVER», в которой имеется исполняемый файл драйвера.



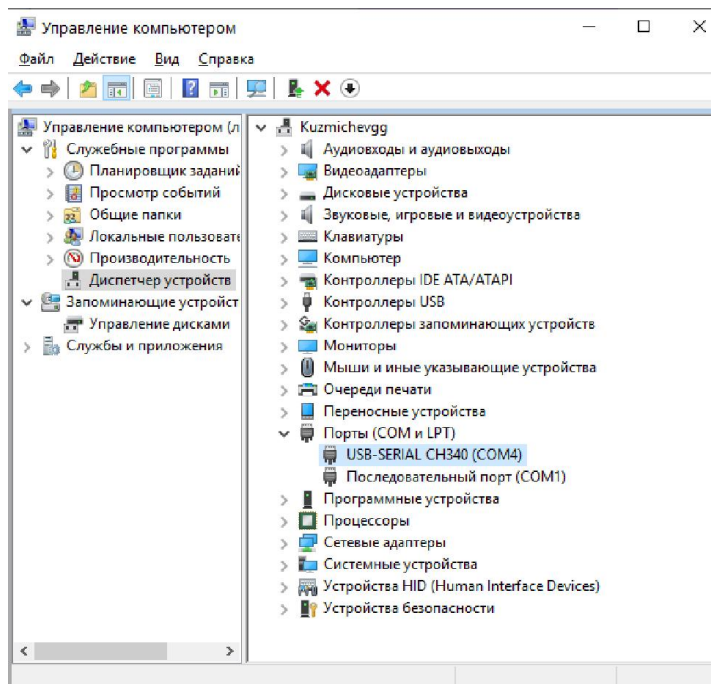
Запускаем его.



В появившемся окне, кликаем на кнопку «INSTALL» и ждем окончания установки драйвера.



После чего закрываем окно приложения и возвращаемся к нашему диспетчеру устройств. В нем видим, что наше неведомое устройство превратилось в последовательный порт.

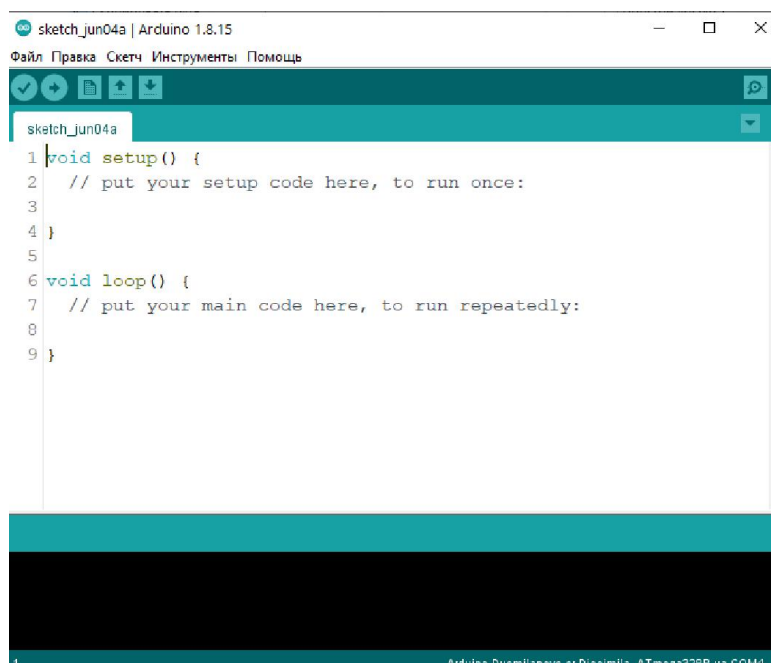


Номер последовательного порта нам понадобится в дальнейшем, для работы с Arduino. Обращаю внимание, что номер порта зависит от того, к какому USB-разъему вы подключите Arduino и соответственно может меняться.

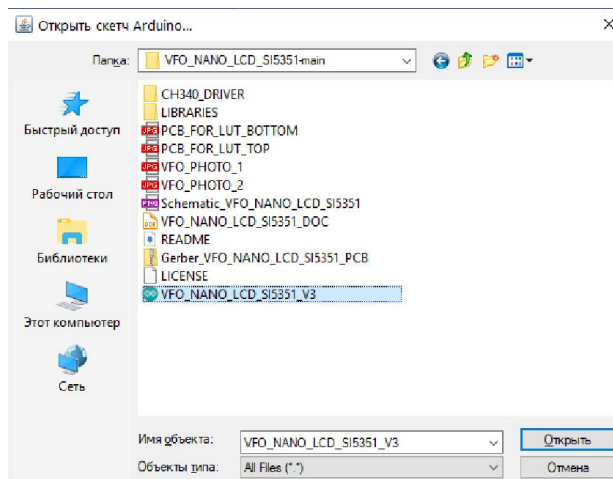
На этом подключение Arduino к компьютеру успешно завершено, осталось записать в микроконтроллер необходимую программу.

Этап 4. Программируем Arduino.

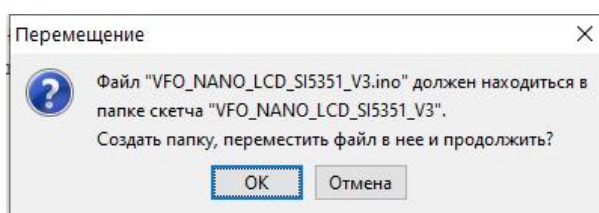
Для начала, нам необходимо запустить среду программирования Arduino. Для этого запускаем программу «Arduino.exe» из папки, куда мы ее разместили, в моем случае это была папка «D:\RADIO\Arduino\».



Через меню «Файл» - «Открыть», находим распакованный нами проект (в папке «Загрузки»), а в нем файл «VFO_NANO_LCD_SI5351_V3.ino».

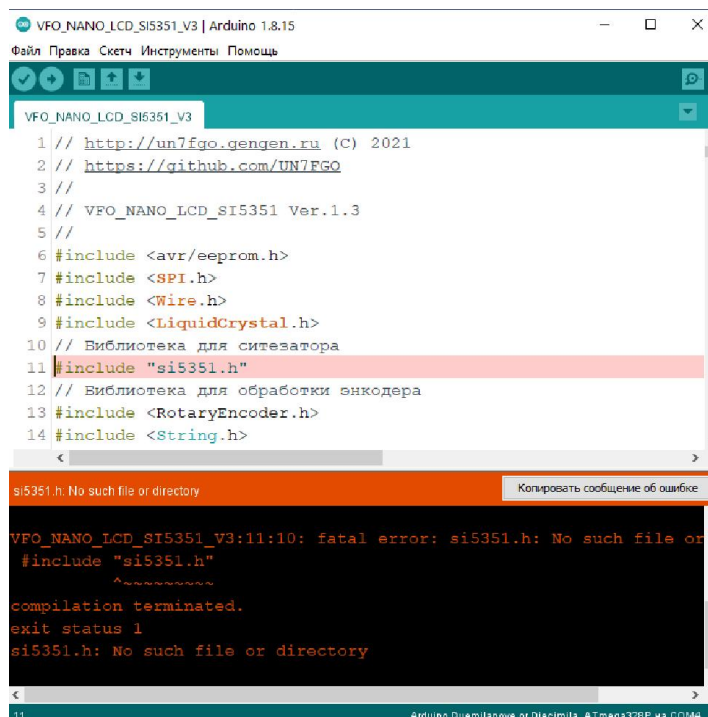


Этот файл выделяется своеобразным значком, похожим на иконку самой среды разработки. При открытии, среда разработки спросит разрешение на перемещение файла.



Нам необходимо будет согласиться, поскольку структура программы в среде разработки для Arduino, подразумевает наличие папки с именем программы, в которой кроме собственно программы, могут находиться различные вспомогательные файлы.

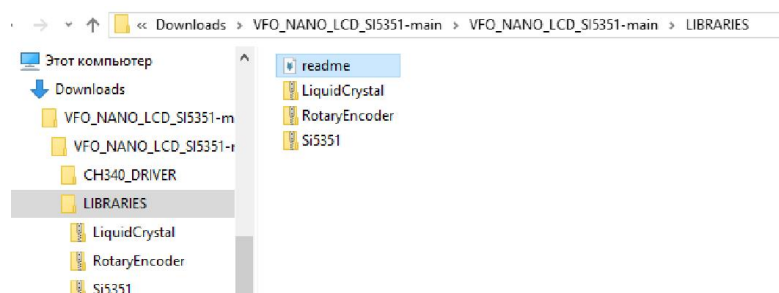
После открытия программы, нажимаем кнопку «Проверить», в левом верхнем углу (круглая кнопка с «галочкой» внутри).



В результате мы видим сообщение об ошибке и выделенную строку со ссылкой на библиотеку синтезатора.

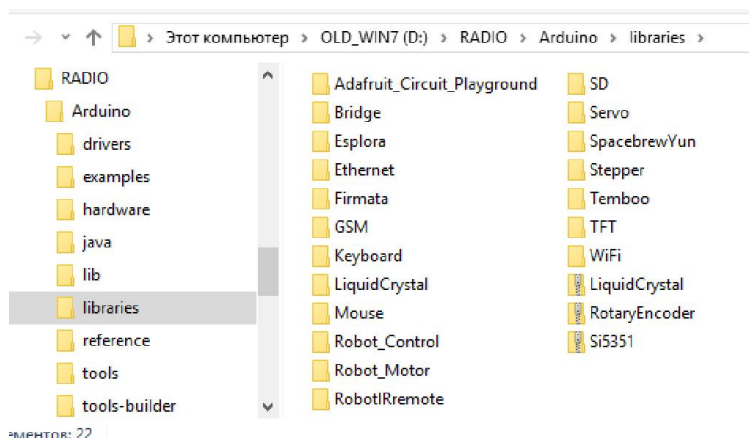
Все правильно. Мы развернули на компьютере только базовую версию среды разработки, которая не знает, с чем и как мы хотим работать. Для работы с различными периферийными устройствами, необходимы библиотеки, в которых описаны методы работы и различные функции, позволяющие делать нам то, что мы хотим. Конечно можно начать искать библиотеки в сети Интернет, но нам это не нужно, все необходимые библиотеки уже есть в скачанном нами архиве.

Переходим к скачанному проекту и находим папку «\VFO_NANO_LCD_SI5351-main\LIBRARIES».



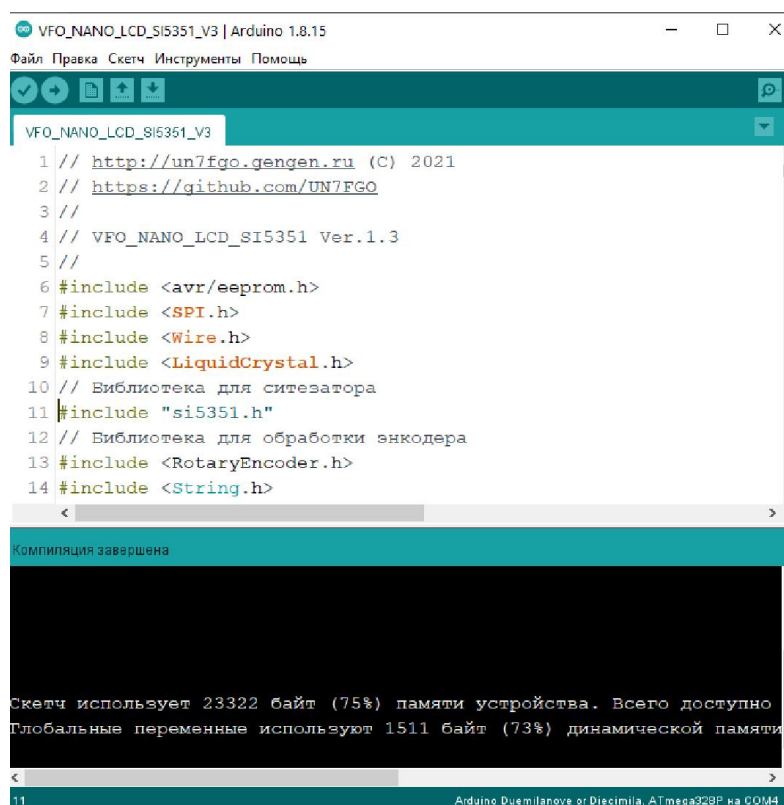
И видим три архива с необходимыми библиотеками.

Копируем эти архивы в папку библиотек установленной среды программирования, в моем случае это «D:\RADIO\Arduino\libraries».



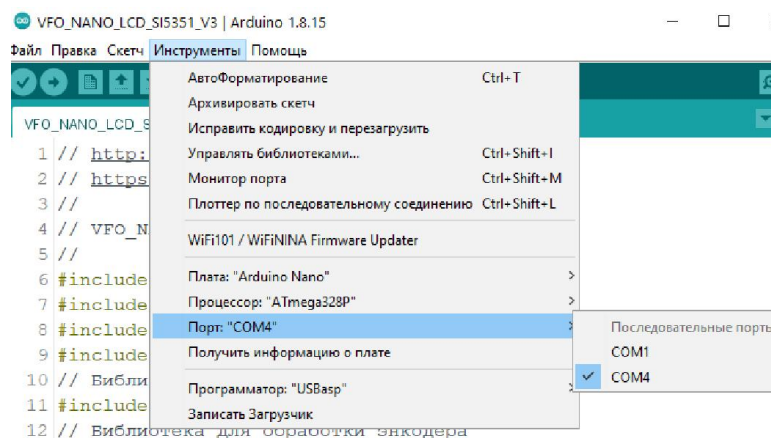
Распаковываем архивы в эту папку, после чего сами архивы можно будет удалить.

Переходим обратно к нашей программе и пробуем еще раз нажать кнопку «Проверить».

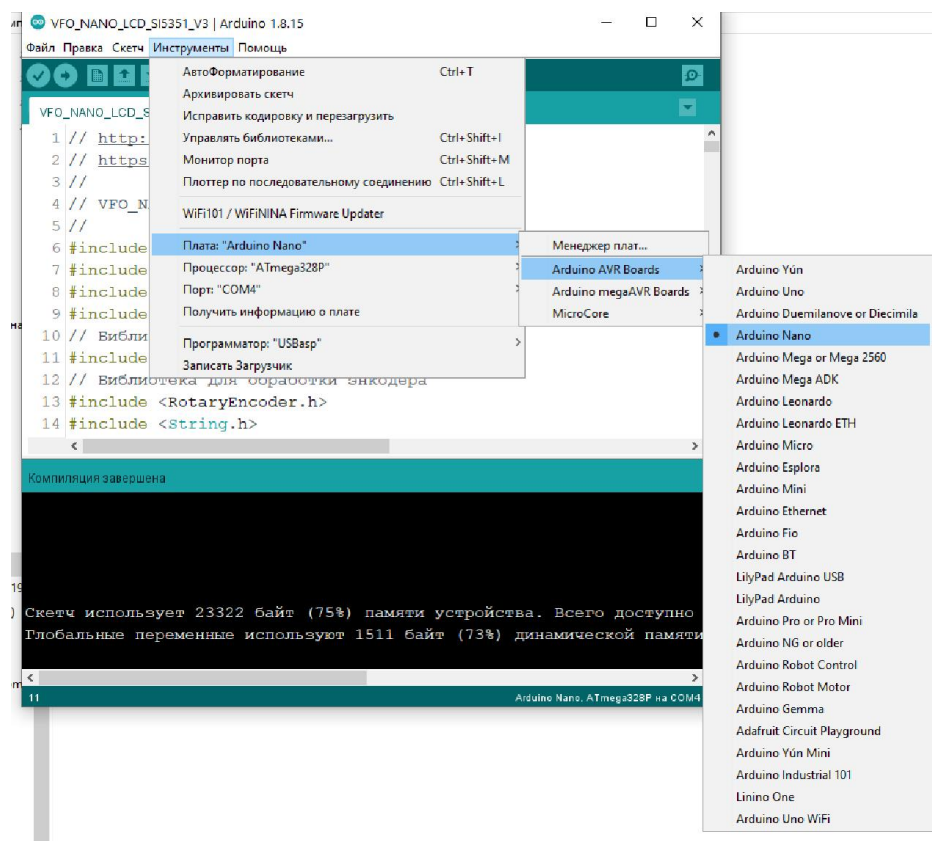


Как видим, в этот раз компиляция программы прошла успешно, и мы не видим сообщений об ошибках. Осталось загрузить программу в сам микроконтроллер.

Для этого, в меню программы, выбираем пункт «Инструменты», раздел «Порт» и выбираем порт, назначенный нашему микроконтроллеру.



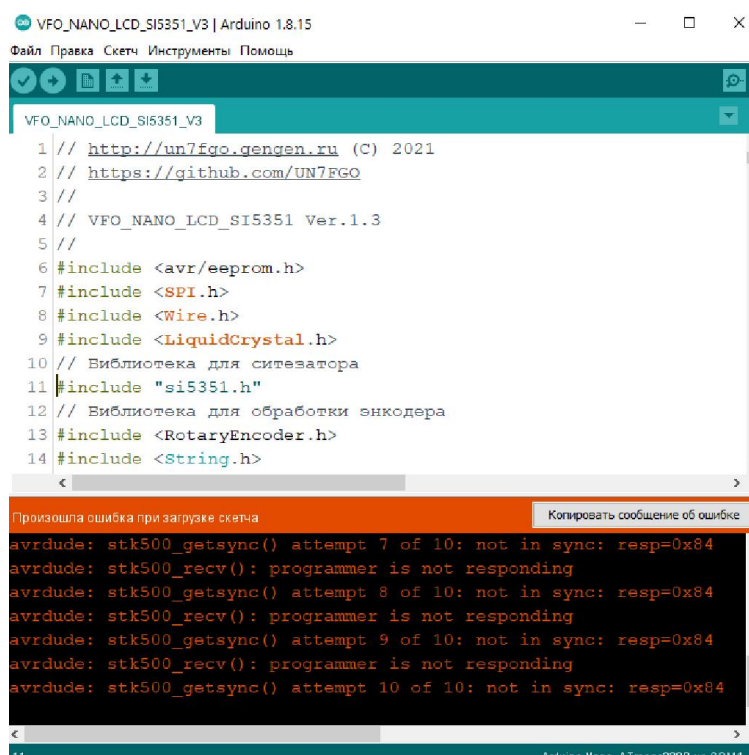
В нашем случае это «COM4». После чего необходимо в разделе «Плата» выбрать тип нашего микроконтроллера.



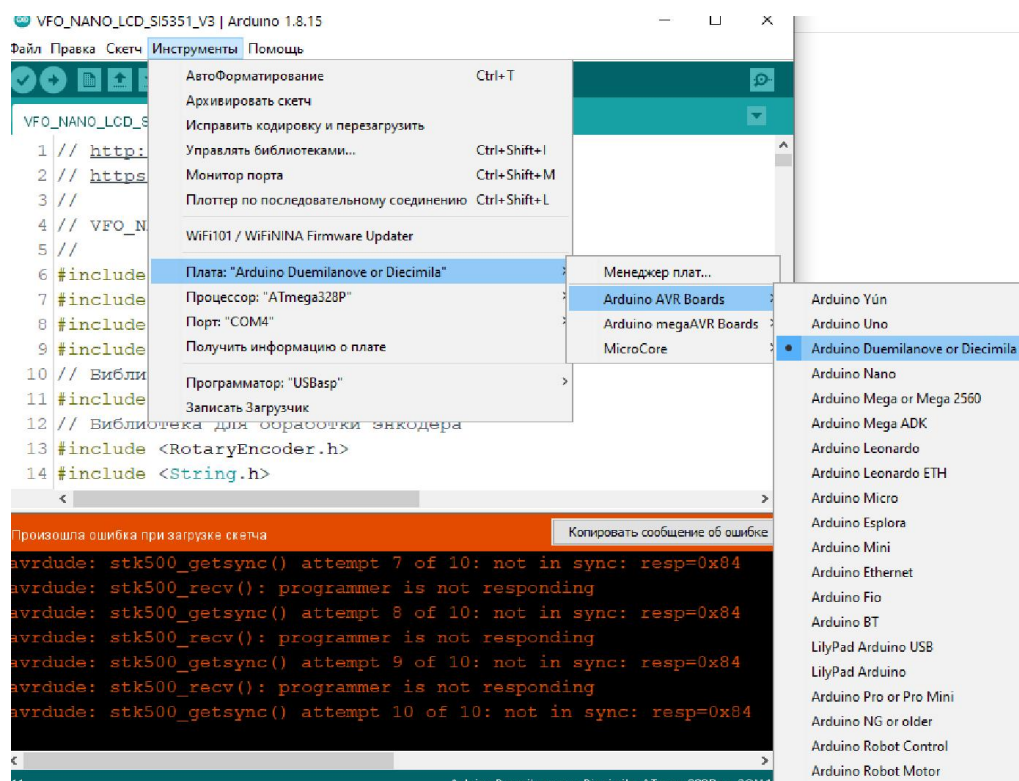
В нашем случае, это «Arduino Nano».

Теперь нажимаем кнопку «Загрузка», это вторая кнопка на панели сверху, на ней изображена стрелка, показывающая направо.

В случае наличия у вас не очень качественного китайского клона Arduino Nano, загрузка скетча не произойдет и после некоторого ожидания, вы вероятно увидите сообщение об ошибке.



Не пугайтесь. Просто производитель загрузил в нашу Arduino Nano, код загрузчика от другого варианта микроконтроллера или просто старый код загрузчика. Для нормальной загрузки программы, нам просто достаточно выбрать другой вид платы, чаще всего это «Arduino Duemilanove or Diecimila».



Не знаю, чем вызвана любовь китайских производителей к загрузчику именно от этой модели, но после выбора ее, процесс загрузки происходит без каких-либо проблем.

```

VFO_NANO_LCD_SI5351_V3 | Arduino 1.8.15
Файл Правка Скетч Инструменты Помощь

VFO_NANO_LCD_SI5351_V3
1 // http://un7fgo.gengen.ru (C) 2021
2 // https://github.com/UN7FGO
3 //
4 // VFO_NANO_LCD_SI5351 Ver.1.3
5 //
6 #include <avr/eeprom.h>
7 #include <SPI.h>
8 #include <Wire.h>
9 #include <LiquidCrystal.h>
10 // Библиотека для синтезатора
11 #include "si5351.h"
12 // Библиотека для обработки энкодера
13 #include <RotaryEncoder.h>
14 #include <String.h>

Загрузка завершена
Скетч использует 23322 байт (75%) памяти устройства. Всего доступно 30
Глобальные переменные используют 1511 байт (73%) динамической памяти,

11 Arduino Duemilanove or Diecimila, ATmega328P на COM4

```

На этом все!

Поздравляю! Вы загрузили первую свою программу в Arduino.

После загрузки программы в микроконтроллер, он перезагружается и начинает самостоятельно выполнять заложенную в него программу.